



# Angular Fundamentals

## Module 1 - Inleiding



Peter Kassenaar  
[info@kassenaar.com](mailto:info@kassenaar.com)

# Peter Kassenaar

- Trainer, author, developer – since 1996
- Specialty: *"Everything JavaScript"*
- JavaScript, ES6, Angular, NodeJS, TypeScript, React, Vue, Phonegap

[www.kassenaar.com](http://www.kassenaar.com)

[info@kassenaar.com](mailto:info@kassenaar.com)

Twitter: [@PeterKassenaar](https://twitter.com/PeterKassenaar)



ING 

OHRA

euricom  
A DIMENSION DATA COMPANY 

s a n o m a

delta lloyd

zenito  
BETERE ZEKERHEID  
VOOR ONDERNEMERS

Atos



OBERON INTERACTIVE

woonbron



the eforum  
FACTORY

Angulartraining.nl Home Training Dates Information Contact

2018 dates now available!



```
const routes: Routes = [
  { path: '', redirectTo: 'home', pathMatch: 'full' },
  { path: 'home', loadChildren: './home/home.module#HomeModule' },
  { path: 'training', loadChildren: './training/training.module#TrainingModule' },
];

const config: ExtraOptions = {
  enableTracing: false, // this on to for console logging of route events
  preloadingStrategy: PreloadingStrategyNoPreload,
};

@NgModule({
  imports: [RouterModule.forRoot(routes, config)],
  exports: [RouterModule]
})
export class AppRoutingModule {}
```

# World-class Angular training in Dutch and English

Live classrooms - focused on today's developers

LEARN MORE SIGN UP!

[www.angulartraining.nl](http://www.angulartraining.nl)

**Over jullie**



Voorkennis webdevelopment, (mobile/web-) apps?

(Kennis AngularJS 1.x?)

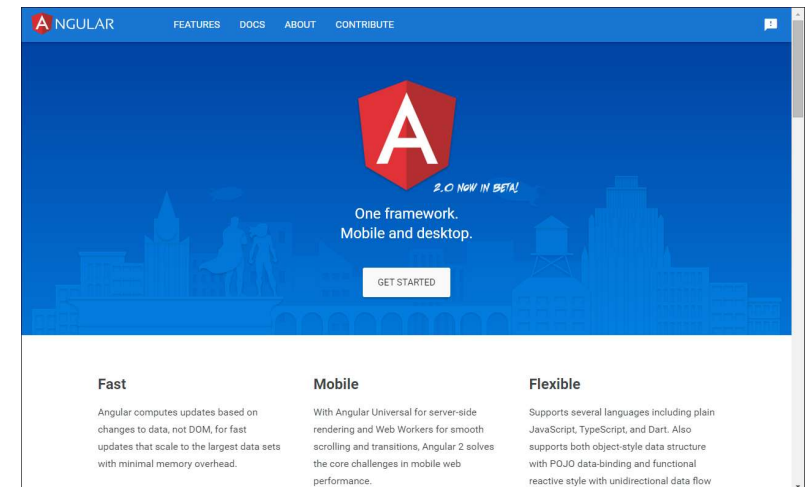
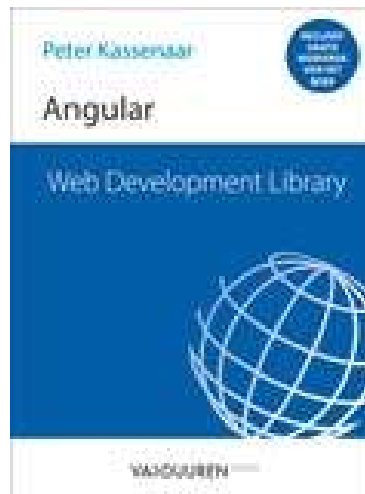
Voorkennis andere (web)talen?

Verwachtingen van de cursus?

Concrete projecten?

# Materialen

Software	(Angular, NodeJS, editor, downloads)
Handouts	(Github - PDF)
Oefeningen	(Github)
Websites	(online)



[angular.io/](https://angular.io/)

# Agenda - 2 dagen

- Introductie & geschiedenis - waarom Angular?
- Hello World in Angular – inzicht in boilerplate-code
- Angular in depth:
  - Components
  - ECMAScript 2015 + TypeScript
  - Data binding
  - Dependency Injection (DI) – more components
  - Services en http, Observables (RxJS)
  - Tree of Components
  - Routing, Forms
- BEST PRACTICES / STYLE GUIDE

# Agenda

- Day 1 - Intro
  - Theory - Introductie & geschiedenis - why Angular
  - Hello World in Angular –boilerplate-code
  - Concepts, context & architecture
  - Angular CLI
  - Components
  - Data binding
  - [Services]



# Agenda

- Day 2 – Services & Communication
  - Services & DI
  - HTTP, Live API's, Observables (RxJS)
  - Applications as a tree of components
  - Routing/forms

## 2 Richtlijnen

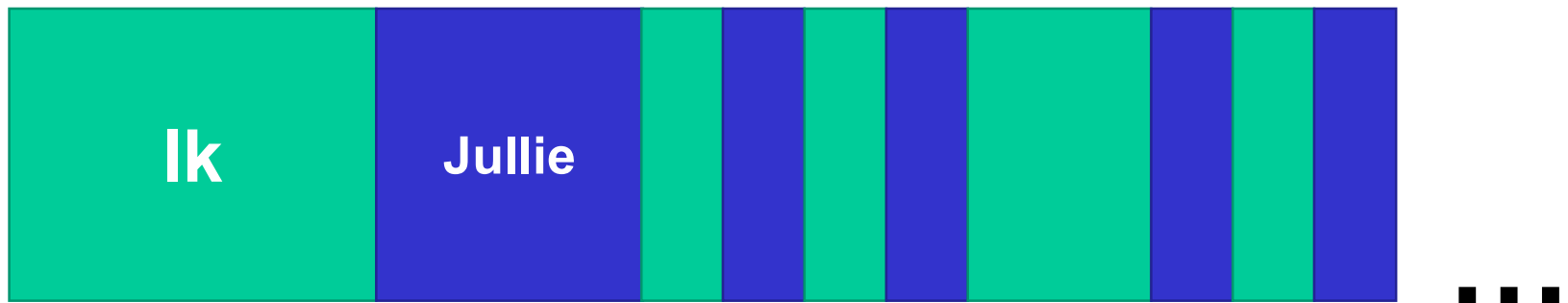
### 1. Oefeningen

- Maar: neem ook vooral zijpaden, experimenteer, lees verder, maak een eigen project, app, website...

### 2. Voorbeeldcode

- Als ondersteuning bij de oefeningen, zie boven
- Work in progress – check de Angular-site!
- [github.com/PeterKassenaar/voorbeeldenAngular2](https://github.com/PeterKassenaar/voorbeeldenAngular2)

# Globale werkwijze



# Vragen?



# Angular vs. The Rest

Differences, similarities, new features

# Addressing the “WHY” question!

WHY, would we want to use a frontend framework.

It is all HTML, CSS and JavaScript right?

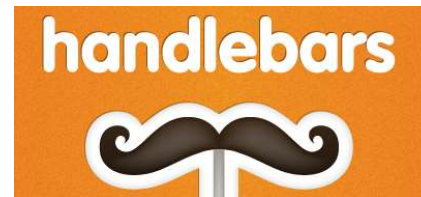
Rethorical question:

*“Do we want to go back  
to the jQuery days?”*

speed,  
consistency, not  
re-inventing the  
wheel, community,  
performance,  
testing....

# Old school web apps

HTML + templates



Data Binding



Routing



DOM-manipulation



Mobile development



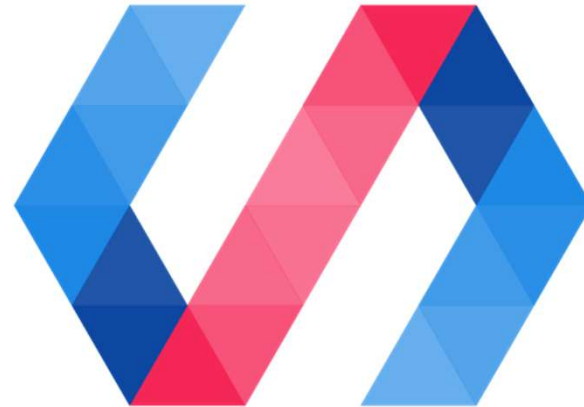
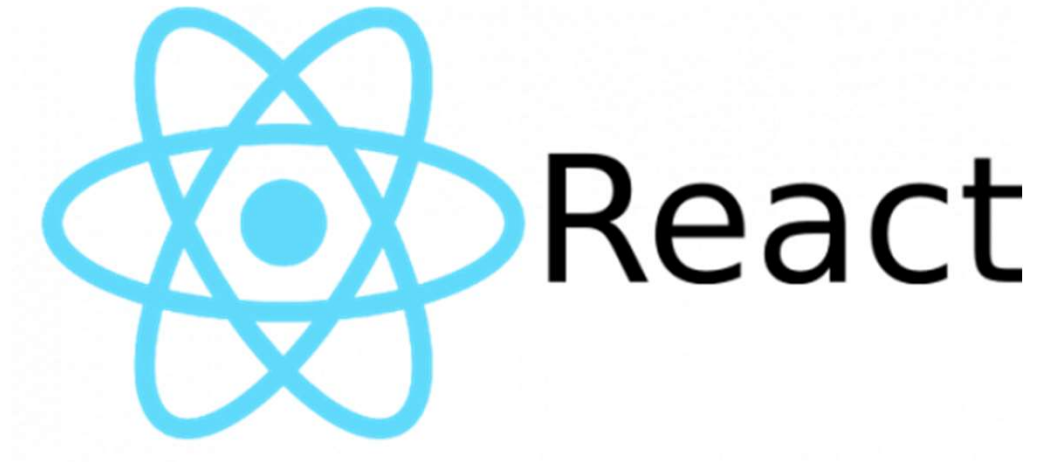
...

# **“The Frankenstein Framework”**





# Front-end Frameworks – the big four

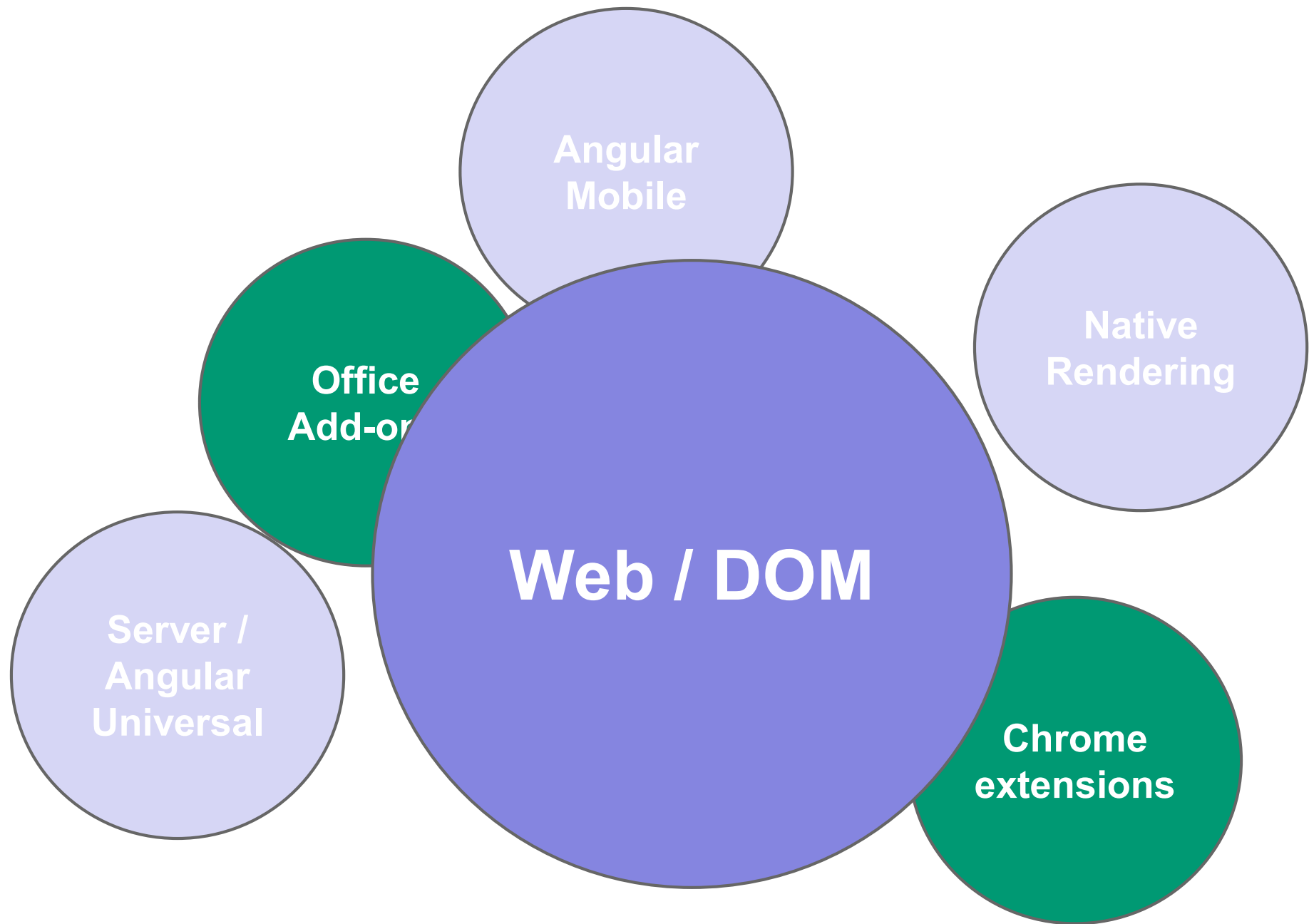




# Platform

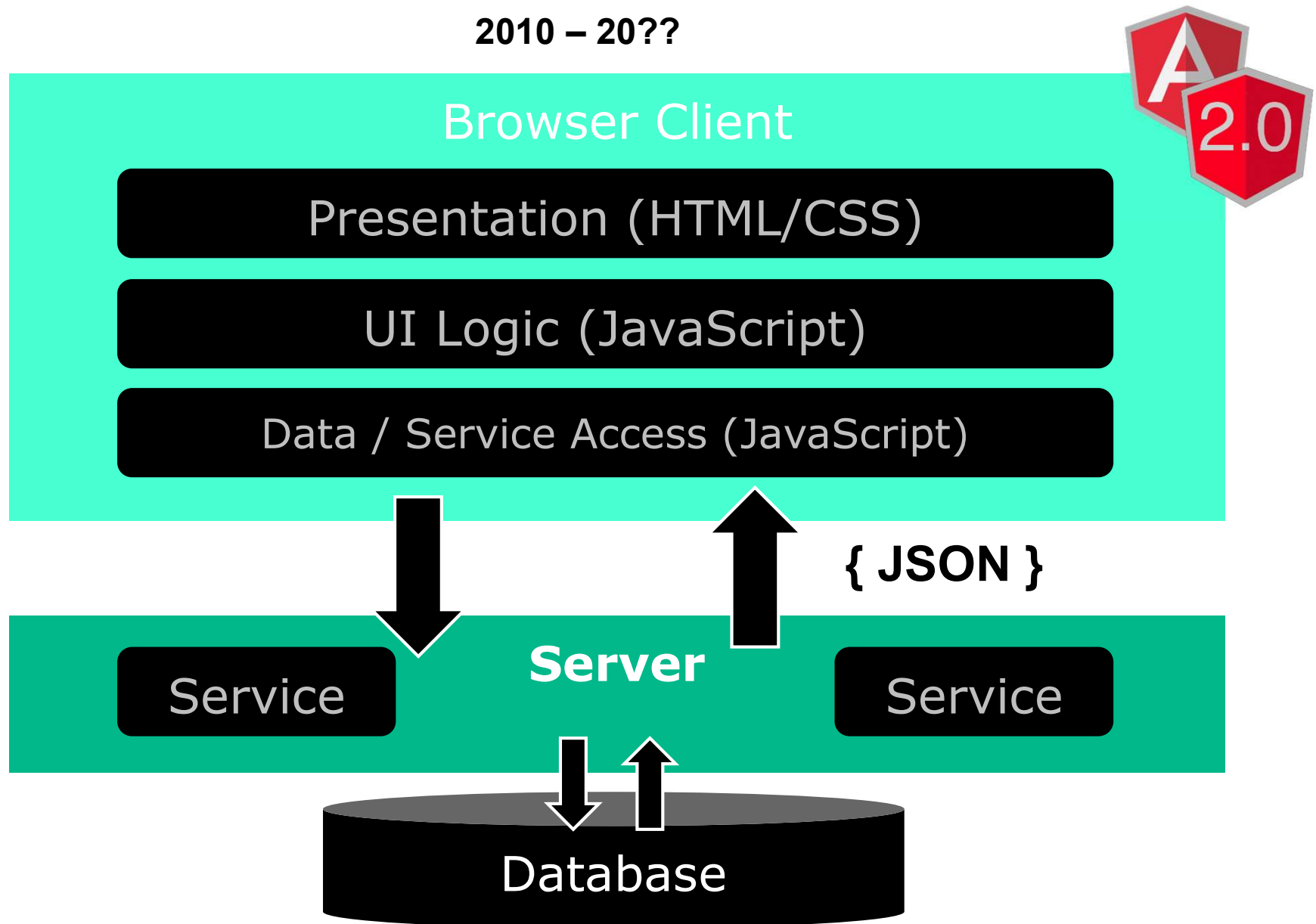
# Platform Features

	Scaffolding	Code completion & Refactoring	Debugging
Tooling	Angular CLI	Language Services	Augury
Libraries	Material 2	Mobile	Universal
	AOT- Compile	Change Detection	Renderer
Core	Components & Dependency Injection	Decorators	Zones



# Single Page Application

2010 – 20??





## Time-based Releases

@IgorMinar, Nov. 9, 2018



Predictable & Continuous  
Evolution

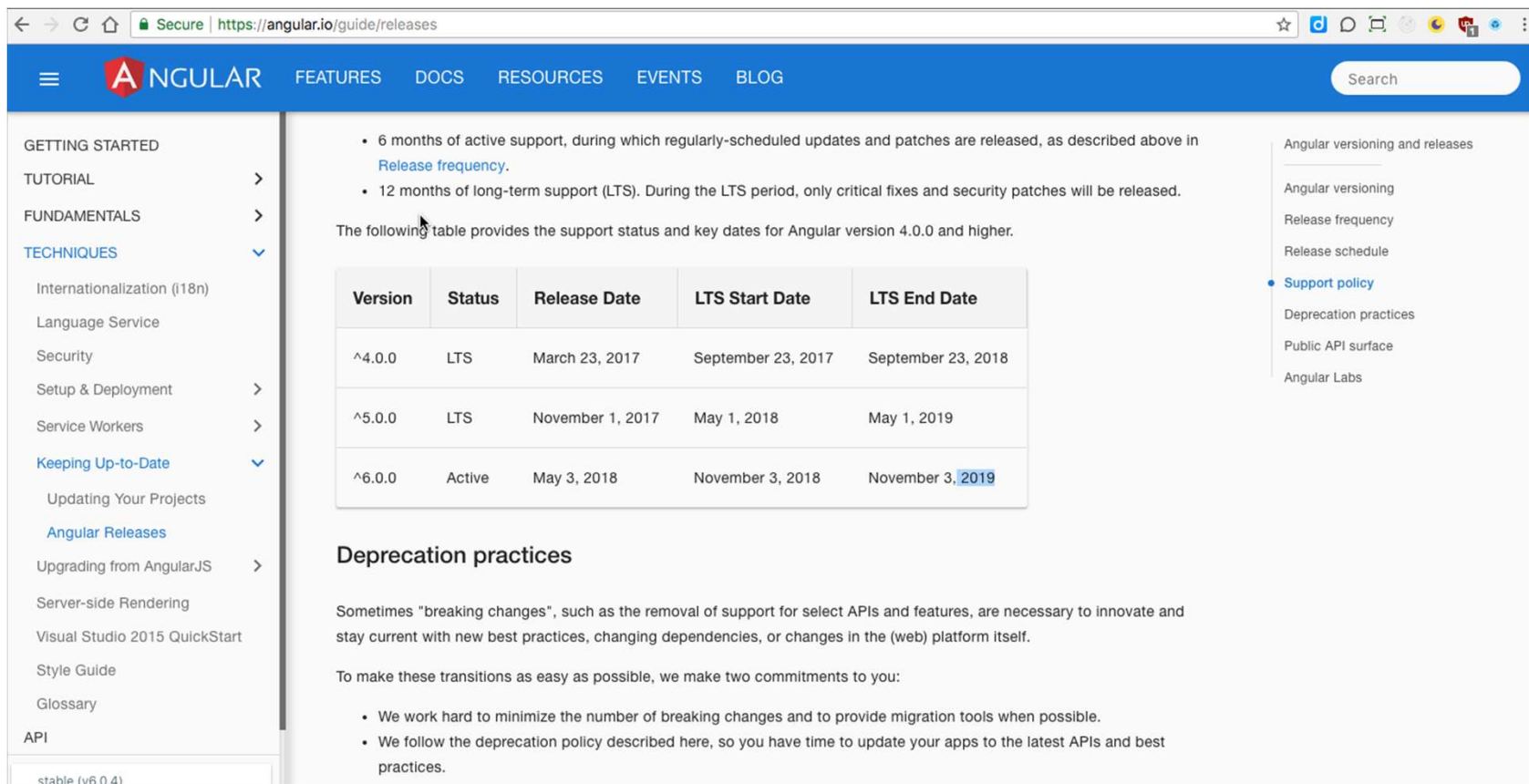


- ✓ Predictability
- ✓ Painless Updates
- ✓ Long Term Support



# Angular Versions en -Long Time Support

→ <https://angular.io/guide/releases>



The screenshot shows the Angular.io website's 'Releases' page. The page is titled 'Angular Versions en -Long Time Support' and includes a navigation menu with links to GETTING STARTED, TUTORIAL, FUNDAMENTALS, TECHNIQUES, and API. The main content area discusses support policies for Angular versions 4.0.0, 5.0.0, and 6.0.0. It mentions that versions 4.0.0 and 5.0.0 have a 12-month LTS period, while version 6.0.0 is currently active. A table provides key dates for each version, including the LTS start and end dates. The table is as follows:

Version	Status	Release Date	LTS Start Date	LTS End Date
^4.0.0	LTS	March 23, 2017	September 23, 2017	September 23, 2018
^5.0.0	LTS	November 1, 2017	May 1, 2018	May 1, 2019
^6.0.0	Active	May 3, 2018	November 3, 2018	November 3, 2019

The page also includes a section on 'Deprecation practices' and a sidebar with links to various Angular resources.

<https://update.angular.io/>

**Angular Update Guide**

Select the options matching your project:

Angular Version

4.0

6.0

App Complexity

Basic

Medium

Advanced

ngUpgrade

☐ I use ngUpgrade

Package Manager

npm

yarn

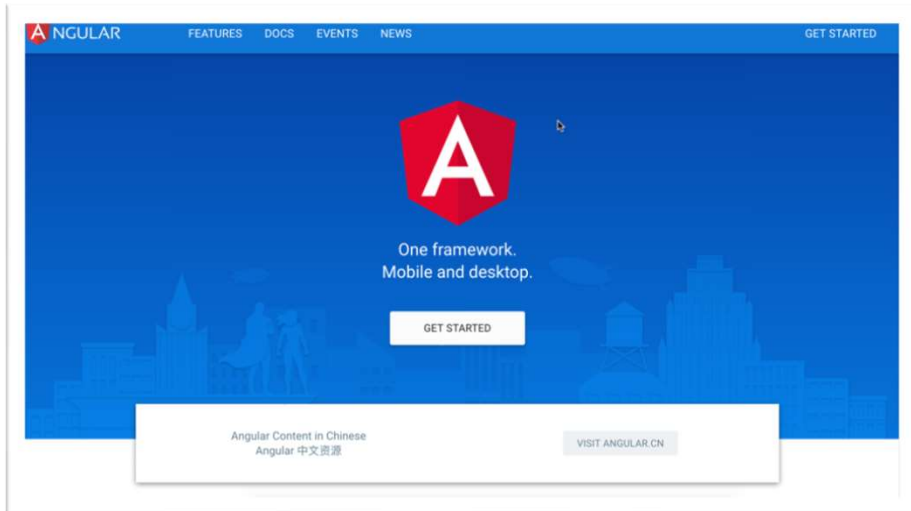
Show me how to update!

**Warning:** We do not recommend moving across multiple major versions.

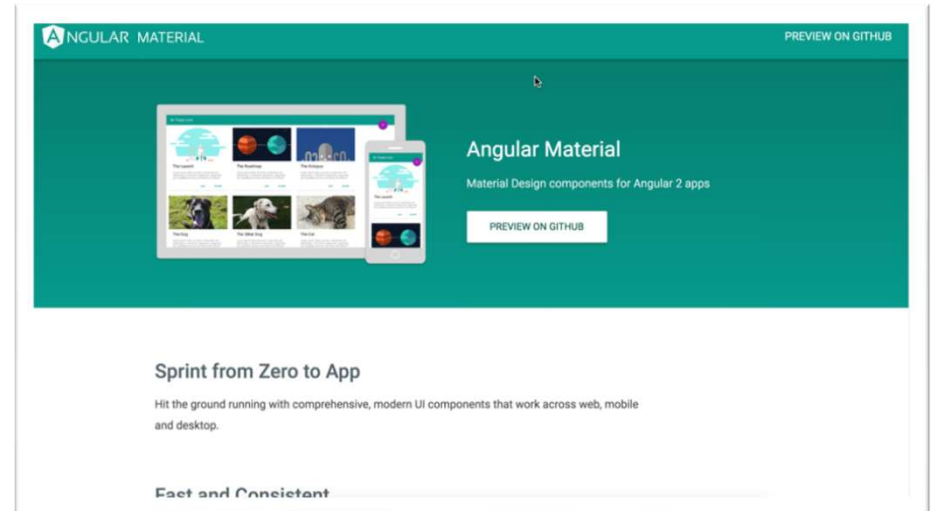
"It's just

*Angular*

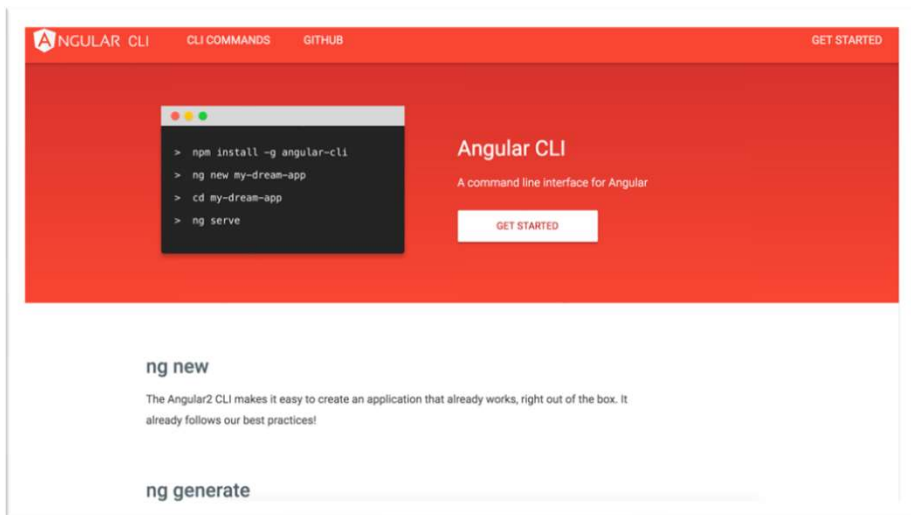
# Angular as a Platform



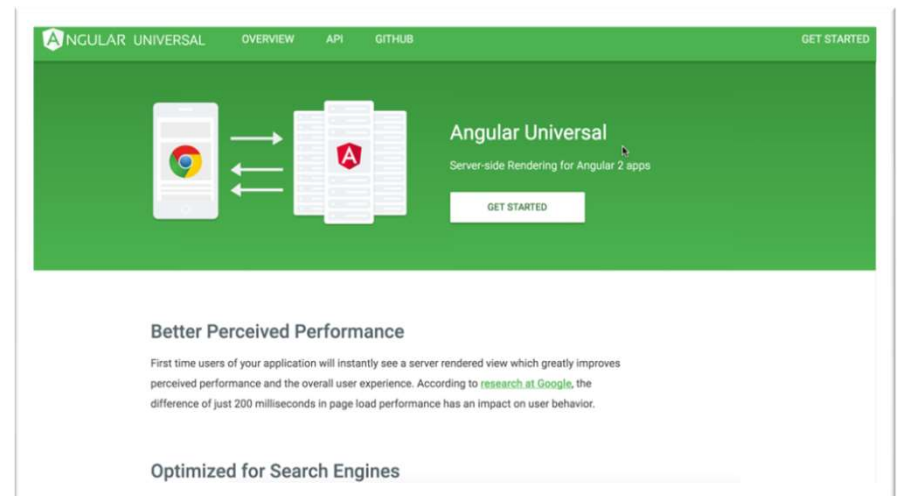
<https://angular.io/>



<https://material.angular.io/>

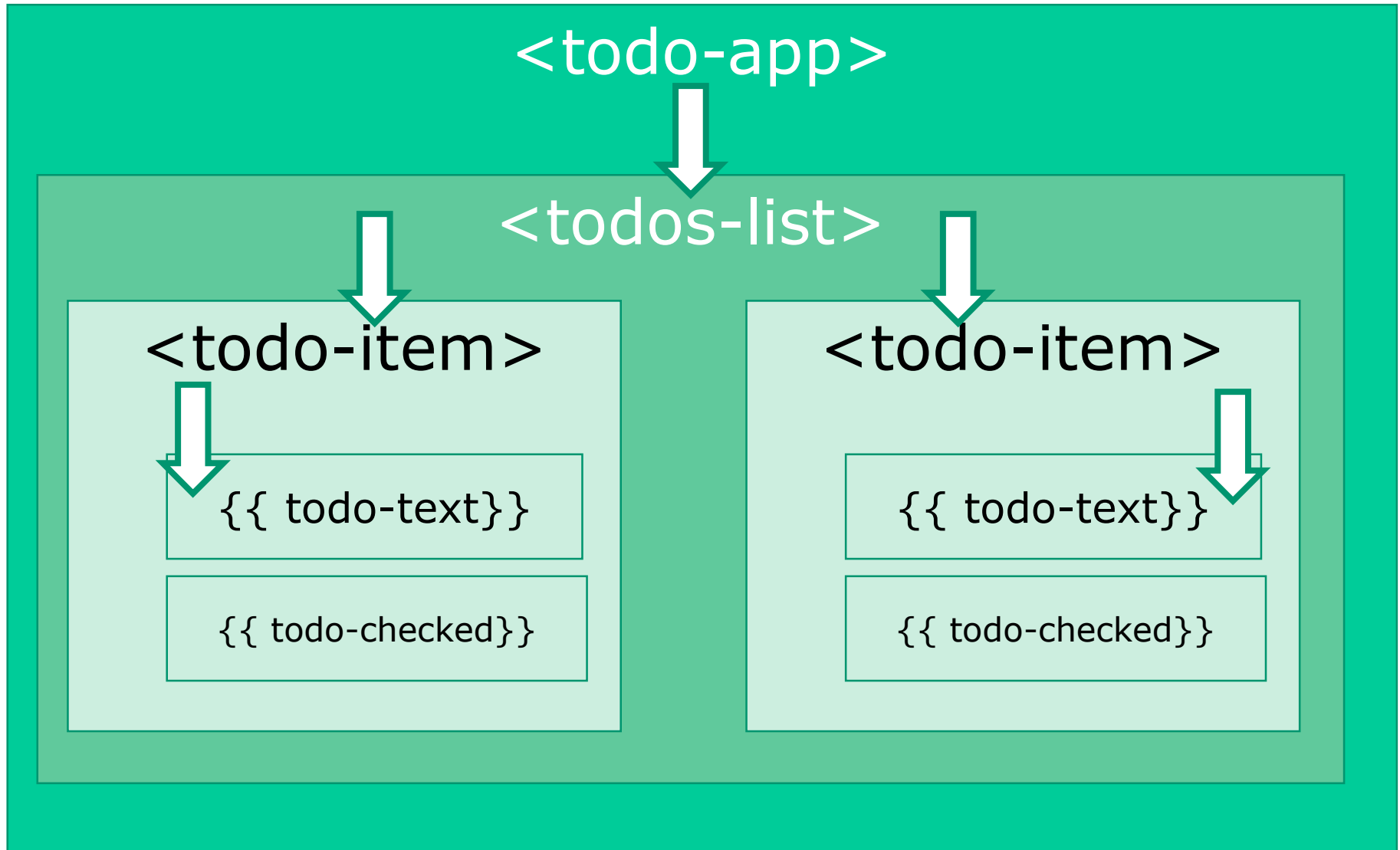


<https://cli.angular.io/>



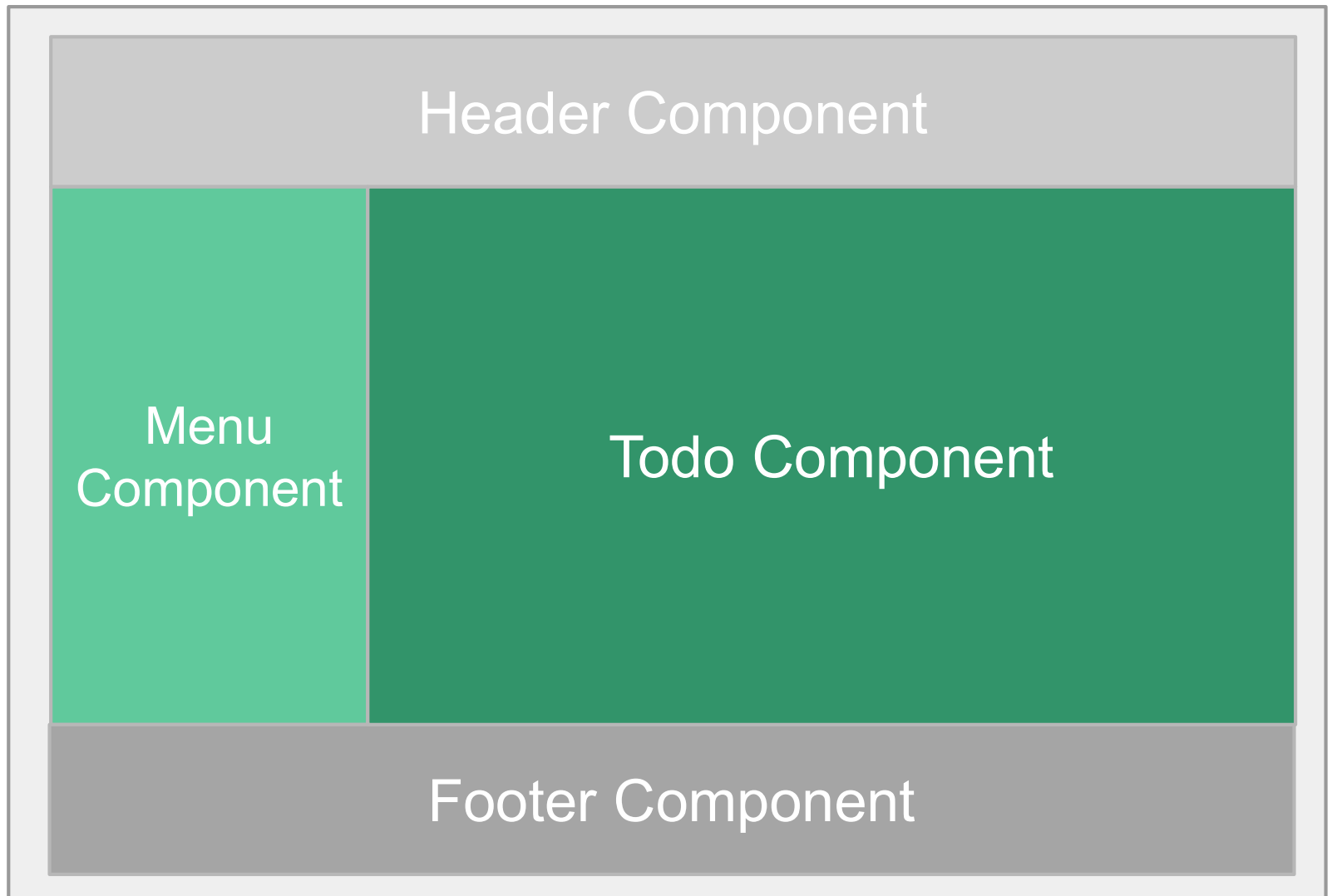
<https://universal.angular.io/>

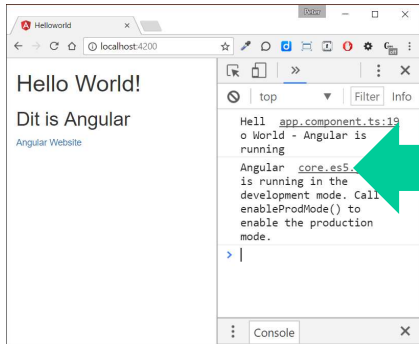
# Angular 2 - components



*"An Angular-app is a tree  
of components"*

## Components – visual representation





**main.ts / bootstrapper**

**ngModule / root module**

**AppComponent**

**Services**

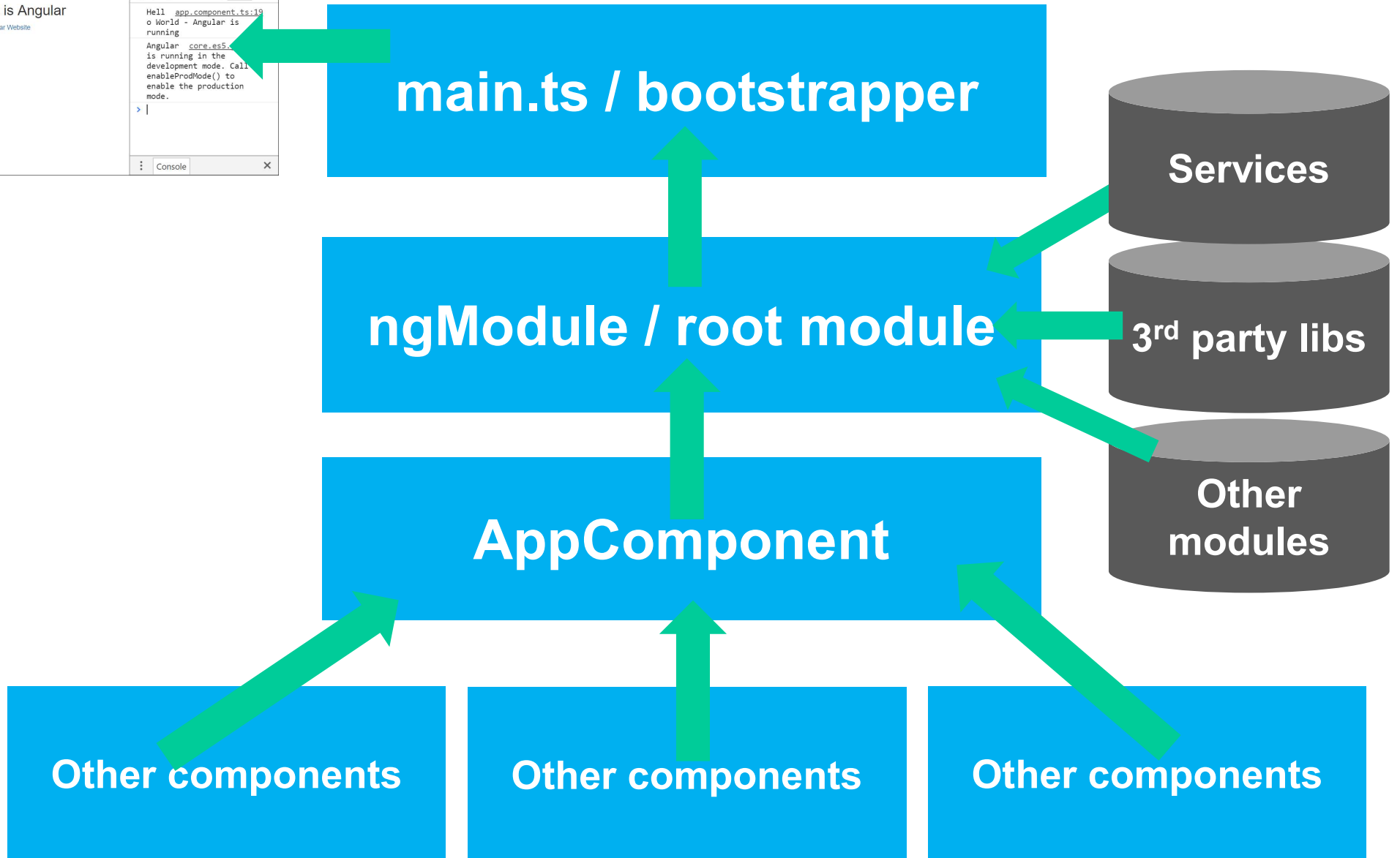
**3<sup>rd</sup> party libs**

**Other modules**

**Other components**

**Other components**

**Other components**







# Let's write some code

Hello World in Angular

## Angular 1:

```
<script src="angular.min.js"></script>
```

# Angular development dependency: NodeJS 10+



The screenshot shows the Node.js website with a dark header containing the Node.js logo and navigation links: HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, NEWS, and FOUNDATION (highlighted in green). Below the header, the text reads: "Node.js® is a JavaScript runtime built on [Chrome's V8 JavaScript engine](#)." The main section is titled "Download for macOS (x64)" and features two green buttons: "12.13.0 LTS" with the subtext "Recommended For Most Users" and "13.0.1 Current" with the subtext "Latest Features". Below these buttons are links for "Other Downloads", "Changelog", and "API Docs" for both versions. Further down, it says "Or have a look at the [Long Term Support \(LTS\) schedule](#)." and "Sign up for [Node.js Everywhere](#), the official Node.js Monthly Newsletter." The footer contains the Linux Foundation logo and text: "COLLABORATIVE PROJECTS", "© Node.js Foundation. All Rights Reserved. Portions of this site originally © Joyent.", "Node.js is a trademark of Joyent, Inc. and is used with its permission. Please review the Trademark Guidelines of the Node.js Foundation.", "Linux Foundation is a registered trademark of The Linux Foundation.", and "Linux is a registered trademark of Linus Torvalds." There are also links for "Report Node.js issue", "Report website issue", and "Get Help".

node

HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | NEWS | FOUNDATION

Node.js® is a JavaScript runtime built on [Chrome's V8 JavaScript engine](#).

Download for macOS (x64)

**12.13.0 LTS**  
Recommended For Most Users

**13.0.1 Current**  
Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#)   [Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the [Long Term Support \(LTS\) schedule](#).

Sign up for [Node.js Everywhere](#), the official Node.js Monthly Newsletter.

 **LINUX FOUNDATION** COLLABORATIVE PROJECTS

[Report Node.js issue](#) | [Report website issue](#) | [Get Help](#)

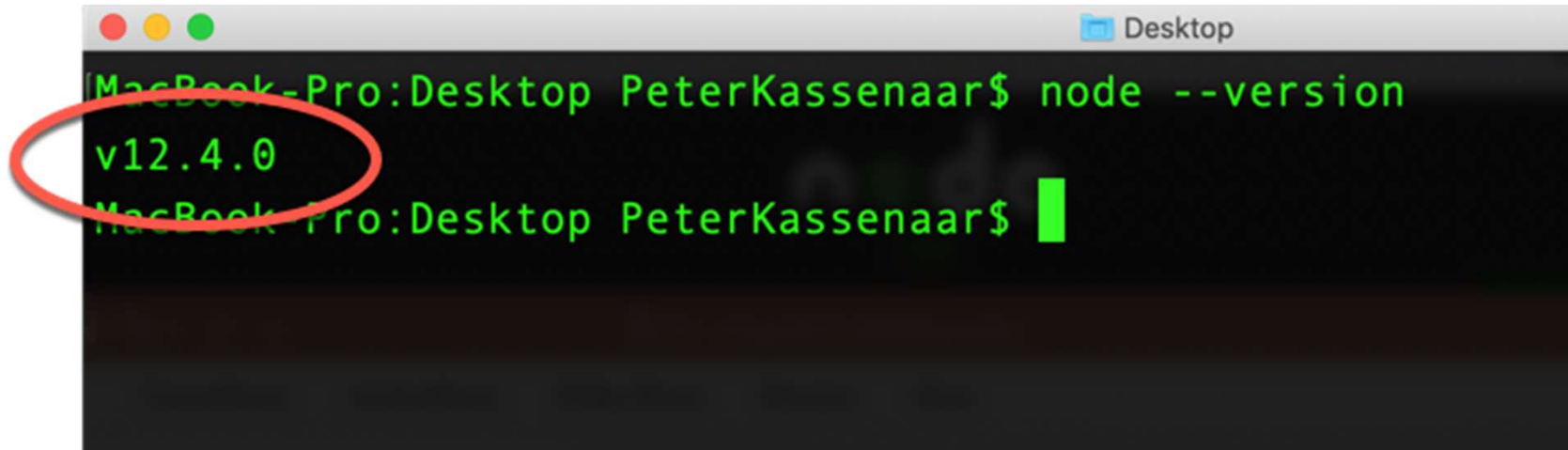
© Node.js Foundation. All Rights Reserved. Portions of this site originally © Joyent.

Node.js is a trademark of Joyent, Inc. and is used with its permission. Please review the Trademark Guidelines of the Node.js Foundation.

Linux Foundation is a registered trademark of The Linux Foundation.

Linux is a registered trademark of Linus Torvalds.

# Node – check your version

A screenshot of a macOS terminal window titled "Desktop". The terminal shows a command prompt "MacBook-Pro:Desktop PeterKassenaar\$" followed by the command "node --version". The output "v12.4.0" is displayed on the next line. A red oval is drawn around the output "v12.4.0". The prompt "MacBook-Pro:Desktop PeterKassenaar\$" is followed by a green cursor bar.

```
MacBook-Pro:Desktop PeterKassenaar$ node --version  
v12.4.0  
MacBook-Pro:Desktop PeterKassenaar$
```

# Mini workshop

- Download or clone <https://github.com/PeterKassenaar/voorbeeldenAngular2>

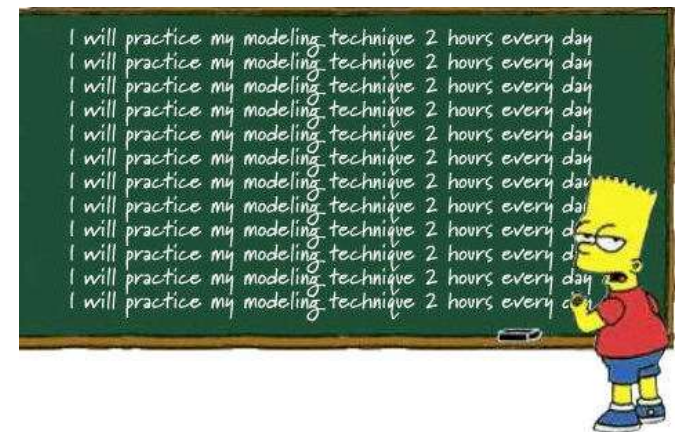
```
cd examples
```

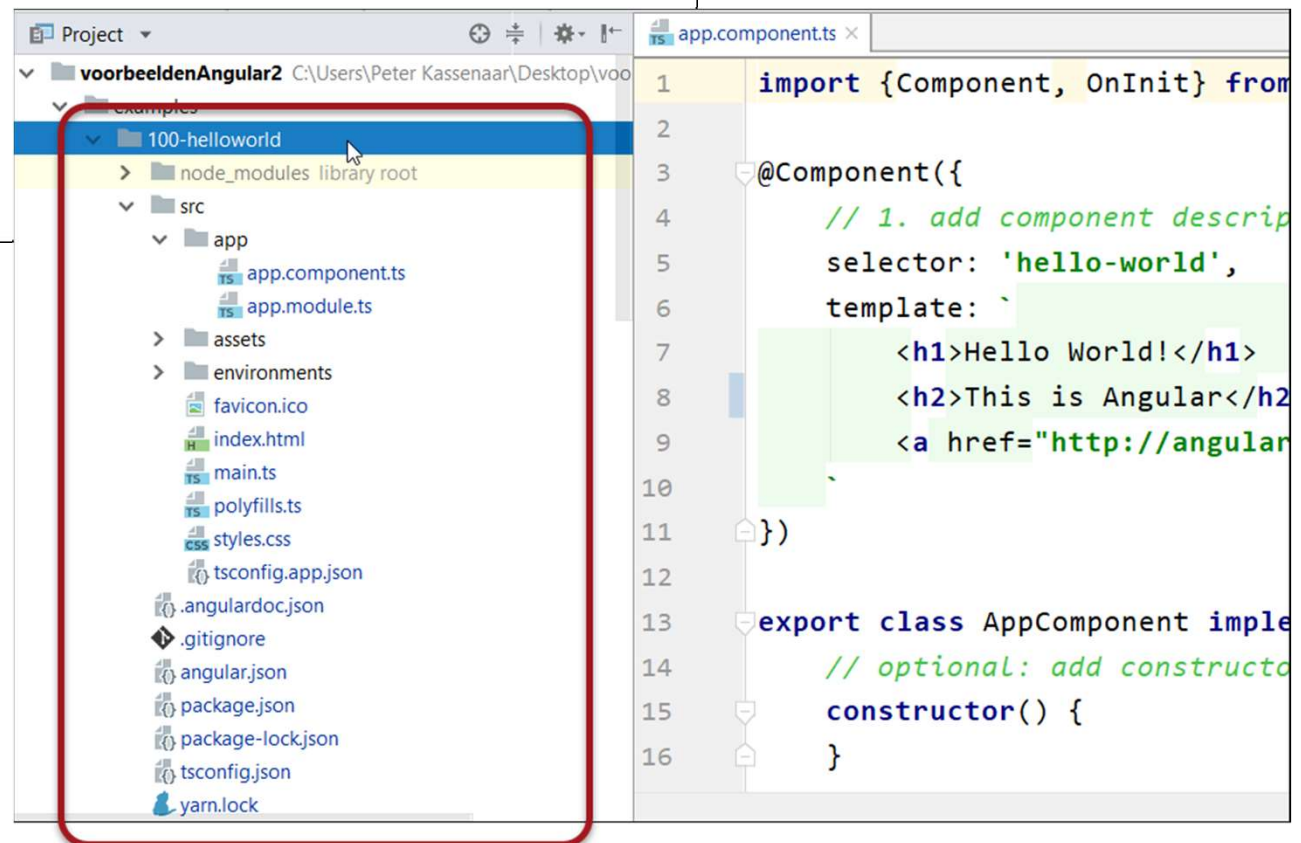
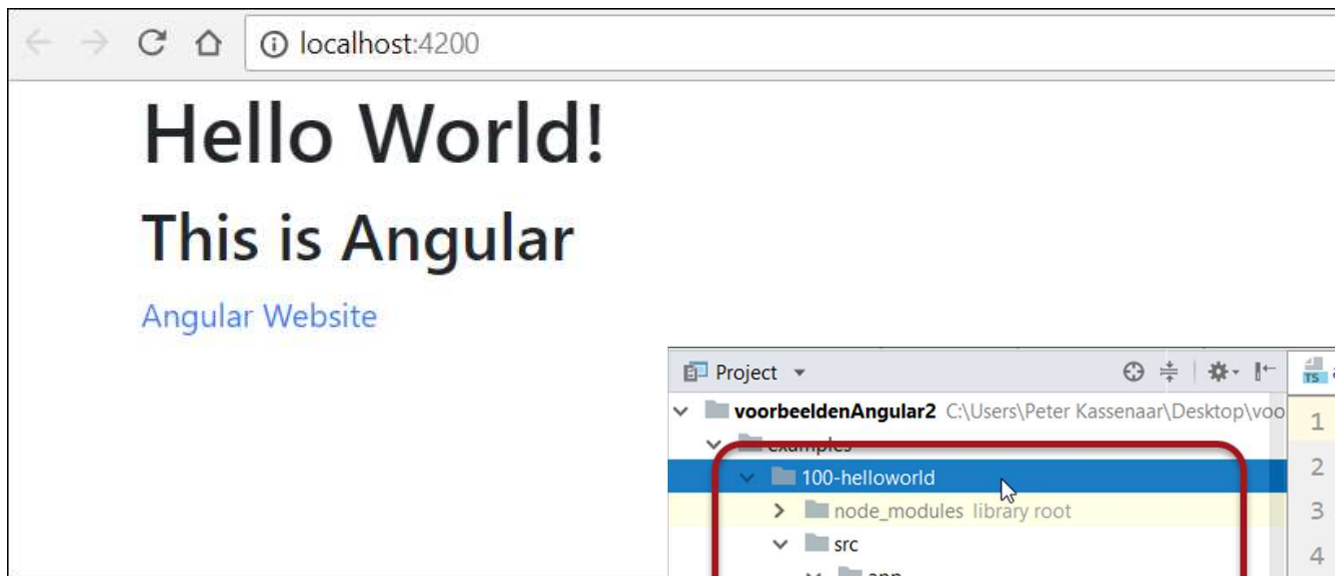
```
cd 100-helloworld
```

```
npm install
```

```
npm start
```

- Go to browser: <http://localhost:4200>





# Boilerplate code for Hello World

## Steps

1. Set up environment, boilerplate & libraries
  - Important configuration files
2. Angular Component(s)
3. Angular Module(s): @ngModule()
4. Bootstrap our module
5. Write HTML-pagina (`index.html`)



# Boilerplate files #1 - package.json

```
{
  "name": "hello-angular",
  "description": "Voorbeeldproject bij de training Angular (C) - info@kassenaar.com",
  "version": "0.0.1",
  "license": "MIT",
  "scripts": {
    "ng": "ng",
    "start": "ng serve",
    "build": "ng build",
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "6.0.0",
    "@angular/common": "6.0.0",
    "@angular/compiler": "6.0.0",
    "@angular/core": "6.0.0",
    "@angular/forms": "6.0.0",
    "rxjs": "^6.1.0",
    "zone.js": "^0.8.26"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "~0.6.0",
    "@angular/cli": "6.0.0",
    "typescript": "2.7.2"
  },
  "author": "Peter Kassenaar <info@kassenaar.com>"
}
```



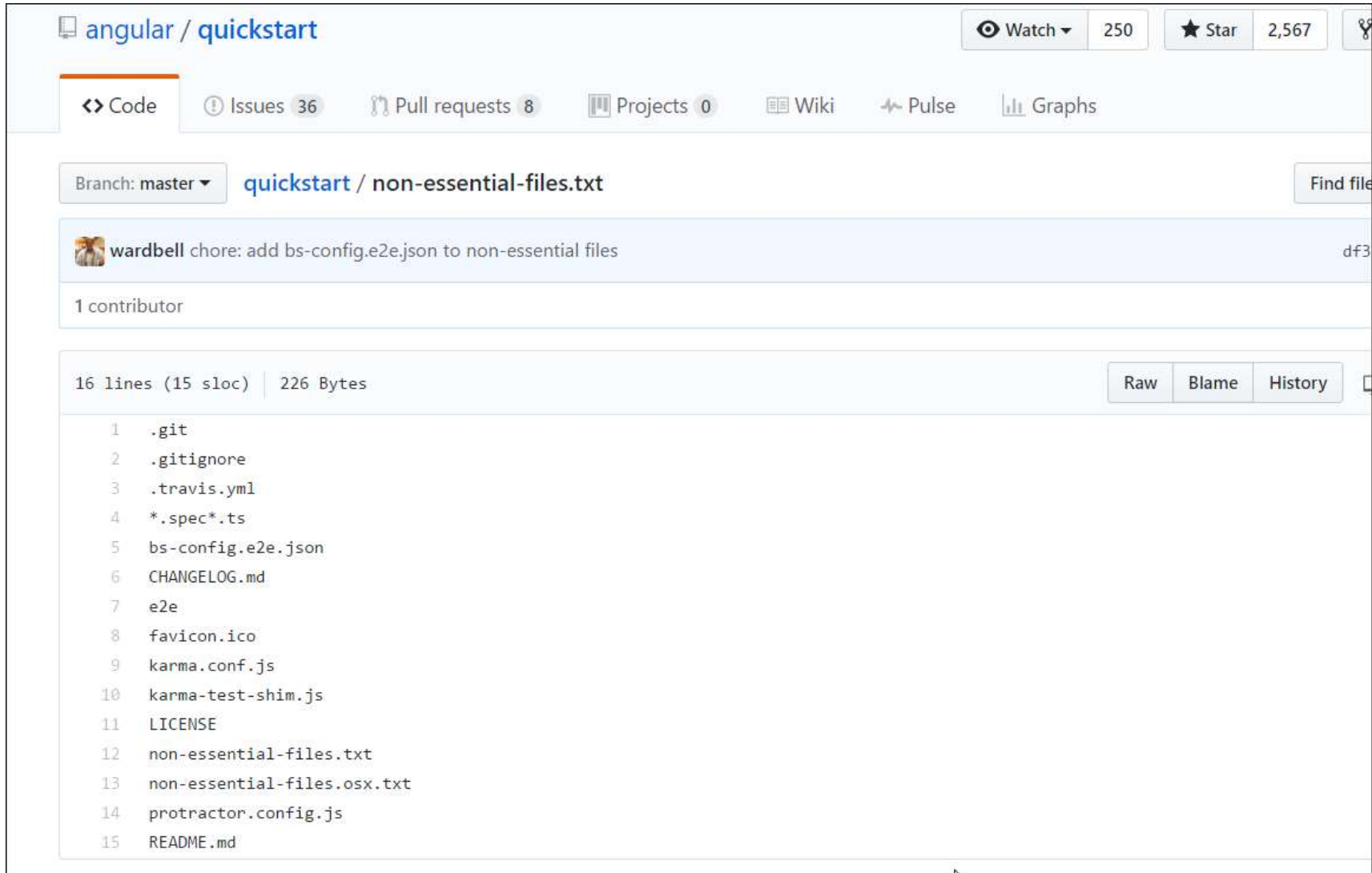
## Boilerplate files #2 - tsconfig.json

```
{
  "compileOnSave" : false,
  "compilerOptions": {
    "outDir"           : "./dist/out-tsc",
    "baseUrl"          : "src",
    "sourceMap"         : true,
    "declaration"       : false,
    "moduleResolution"  : "node",
    "emitDecoratorMetadata" : true,
    "experimentalDecorators": true,
    "target"            : "es5",
    "typeRoots"         : [
      "node_modules/@types"
    ],
    "lib"               : [
      "es2016",
      "dom"
    ]
  }
}
```

## Boilerplate files #3 – angular.json

```
{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "version": 1,
  "newProjectRoot": "projects",
  "projects": {
    "helloworld": {
      "root": "",
      "sourceRoot": "src",
      "projectType": "application",
      "architect": {
        "build": {
          "builder": "@angular-devkit/build-angular:browser",
          "options": {
            "outputPath": "dist",
            "index": "src/index.html",
            "main": "src/main.ts",
            "tsConfig": "src/tsconfig.app.json",
            ...
          }
        }
      }
    }
  }
}
```

# "Nice to have" - non-essential files



The screenshot shows the GitHub interface for the 'angular/quickstart' repository. At the top, the repository name is displayed with 'Watch' (250), 'Star' (2,567), and a search icon. Below this is a navigation bar with links for 'Code', 'Issues' (36), 'Pull requests' (8), 'Projects' (0), 'Wiki', 'Pulse', and 'Graphs'. The main content area shows the file 'quickstart / non-essential-files.txt' on the 'master' branch. A commit by 'wardbell' is highlighted, with the message 'chore: add bs-config.e2e.json to non-essential files'. Below the commit, it says '1 contributor'. The file details show '16 lines (15 sloc)' and '226 Bytes'. On the right, there are buttons for 'Raw', 'Blame', and 'History'. The file content is listed as follows:

```
1 .git
2 .gitignore
3 .travis.yml
4 *.spec*.ts
5 bs-config.e2e.json
6 CHANGELOG.md
7 e2e
8 favicon.ico
9 karma.conf.js
10 karma-test-shim.js
11 LICENSE
12 non-essential-files.txt
13 non-essential-files.osx.txt
14 protractor.config.js
15 README.md
```

<https://github.com/angular/quickstart/blob/master/non-essential-files.txt>

## Step 2 – Component

Convention - components in directory `/src/app`

Or: edit in `angular.json`

Filename: `src/app/app.component.ts`

```
import {Component} from '@angular/core';  
@Component({  
  selector: 'hello-world',  
  template: '<h1>Hello Angular</h1>'  
})  
export class AppComponent {  
  
}
```

## Step 3 – @NgModule

Convention - filename: `/src/app.module.ts`

```
// Angular Modules
import {NgModule}      from '@angular/core';
import {BrowserModule} from '@angular/platform-browser';

// Custom Components
import {AppComponent} from './app.component';

// Module declaration
@NgModule({
  imports      : [BrowserModule],
  declarations: [AppComponent],
  bootstrap   : [AppComponent]
})
export class AppModule {
}
```

Root Module of the application

# Some background info on Root Module



<https://johnpapa.net/introducing-angular-modules-root-module/>

## Step 4 - bootstrap component

Best practice: bootstrap app in separate component

Convention: `main.ts`, of `app.main.ts`.

```
import {enableProdMode} from '@angular/core';
import {platformBrowserDynamic} from '@angular/platform-browser-dynamic';

import {AppModule} from '../app/app.module';
import {environment} from '../environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule);
```

# Step 5 – index.html

index.html - simple HTML file - expanded at runtime by WebPack

Header:

```
<html>
```

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <title>Helloworld</title>
```

```
  <base href="/">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
  <link rel="icon" type="image/x-icon" href="favicon.ico">
```

```
</head>
```



# Body van index.html

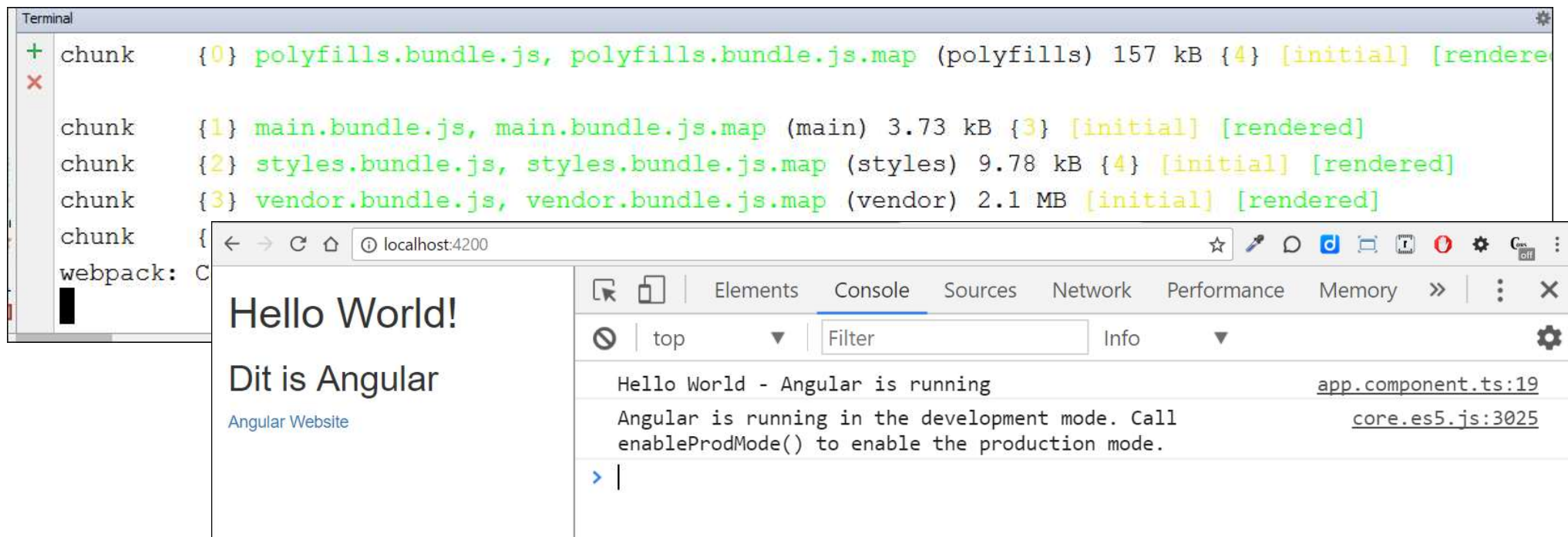
Verwijzing naar de root-component:

```
<body>  
  <hello-world>  
    Bezig met laden...  
  </hello-world>  
</body>
```

# App draaien

`npm start` – draait de scriptopdracht start uit `package.json`.

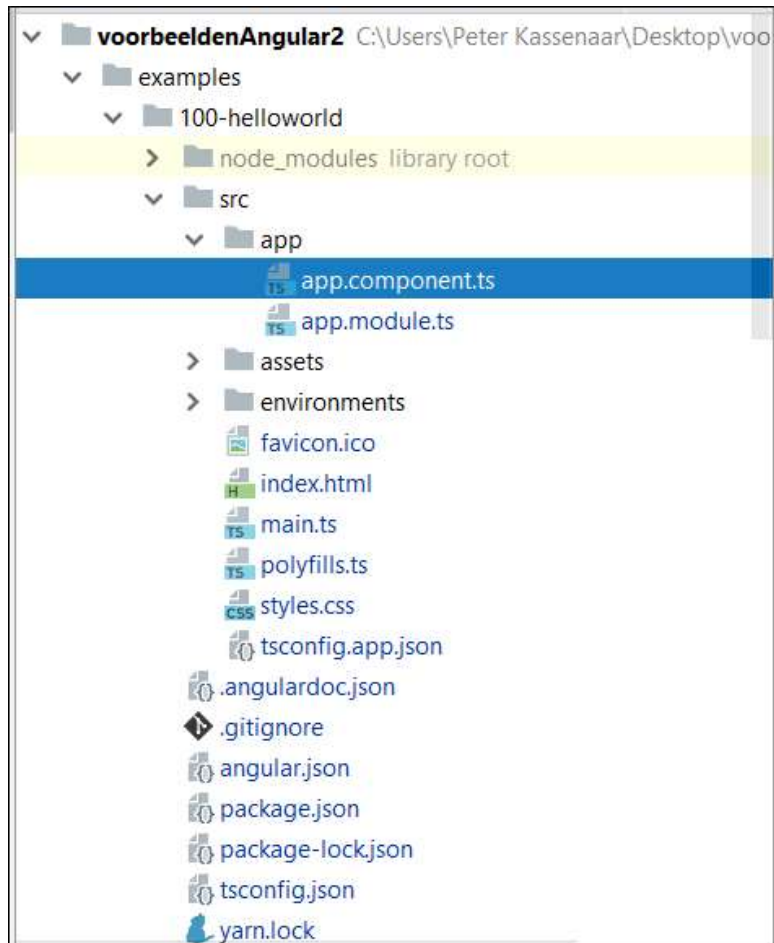
`ng serve` – start globale angular-cli instantie

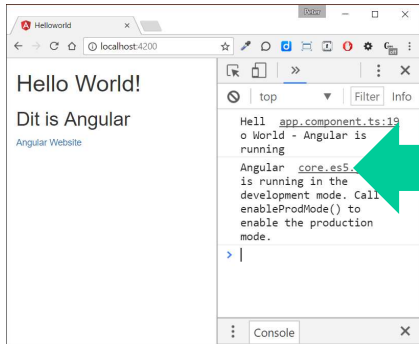


Daarna: wijzigingen aanbrengen in `app.component.ts`

– worden opgepikt door Live Reload

# Basic Project Structure





**main.ts / bootstrapper**

**ngModule / root module**

**AppComponent**

**Services**

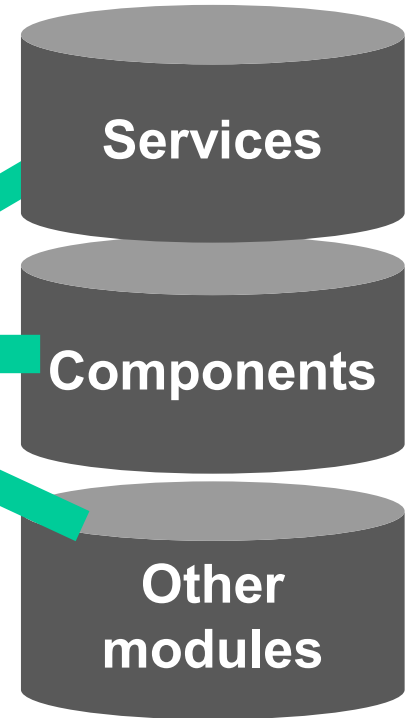
**Components**

**Other  
modules**

**Other components**

**Other components**

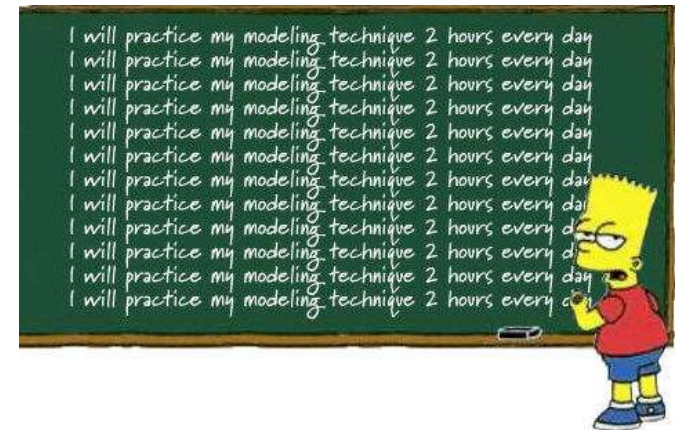
**Other components**



# Checkpoint

- Er is aardig wat boilerplate code nodig om een Angular-app te starten
- Vier stappen
  1. Set up environment, boilerplate & libraries
  2. Schrijf Angular Root Component voor de app
  3. Bootstrap de component
  4. Schrijf HTML-pagina (`index.html`)
- Daarna: app gaan uitbreiden
- Oefening 1a), 1b), 1c), 1d)

## Oefening....



# Assets

[github.com/PeterKassenaar/voorbeeldenAngular2](https://github.com/PeterKassenaar/voorbeeldenAngular2)

Oefeningen en meer voorbeeldcode



# Angular CLI

Snel nieuwe projecten instellen via de command line

# Angular-CLI to the rescue

- Het *is* mogelijk nieuwe Angular-projecten from scratch te starten.
- Met de CLI is eenvoudiger.
- CLI-options:
  - Scaffolding
  - Generating
  - Testing
  - Building
  - AOT-Compiling
  - ...



# Scaffolding - Angular CLI

Projecten, componenten, routes en meer definiëren  
vanaf de command line

<https://github.com/angular/angular-cli>

en

<https://cli.angular.io/>

typings.json chore(): add typings files to assist devs 4 months ago

README.md

## Angular-CLI

chat on gitter

build passing dependencies up to date devDependencies up to date npm v1.0.0-beta.5

Prototype of a CLI for Angular 2 applications based on the [ember-cli](#) project.

### Note

This project is very much still a work in progress.

We still have a long way before getting out of our alpha stage. If you wish to collaborate while the project is still young, check out [our issue list](#).

### Prerequisites

The generated project has dependencies that require **Node 4 or greater**.

### Table of Contents

- [Installation](#)
- [Usage](#)
- [Generating a New Project](#)
- [Generating Components, Directives, Pipes and Services](#)
- [Generating a Route](#)
- [Creating a Build](#)
- [Running Unit Tests](#)
- [Running End-to-End Tests](#)
- [Deploying the App via GitHub Pages](#)
- [Linting and formatting code](#)
- [Support for offline applications](#)
- [Commands autocompletion](#)
- [CSS preprocessor integration](#)
- [3rd Party Library Installation](#)

```
npm install -g @angular/cli
```

```
> npm install -g angular-cli  
> ng new my-dream-app  
> cd my-dream-app  
> ng serve
```

## Angular CLI

A command line interface for Angular

GET STARTED

### ng new

The Angular2 CLI makes it easy to create an application that already works, right out of the box. It already follows our best practices!

### ng generate

Generate components, routes, services and pipes with a simple command. The CLI will also create simple test shells for all of these.



CLI can...

```
> ng generate component my-comp
```

NG  
CONF

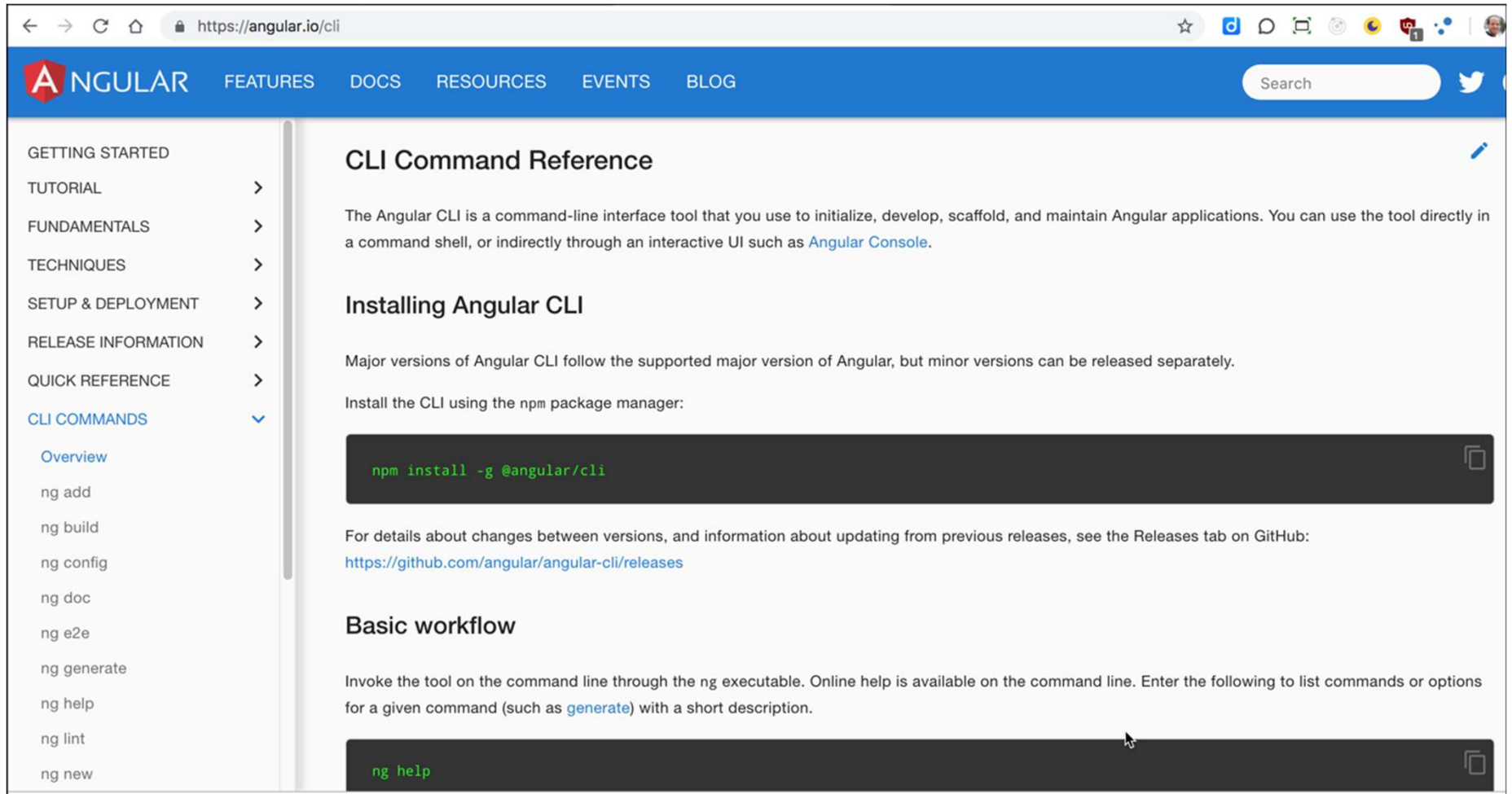
4:50 / 23:39

Learn Clingon - Mike Brocchi

The image shows a YouTube video player interface. The main content area displays a terminal window with the text 'CLI can...' and a command prompt '> ng generate component my-comp'. To the right of the terminal is a small video inset showing a man in a green shirt standing on a stage. Below the terminal is a large 'NG CONF' logo. At the bottom of the video player, there is a progress bar showing '4:50 / 23:39' and standard video controls. Below the video player, the video title 'Learn Clingon - Mike Brocchi' is displayed.

<https://www.youtube.com/watch?v=wHZe6gGI5RY>

# Documentatie - in de Angular Docs



The screenshot shows the Angular CLI documentation page. The browser address bar displays `https://angular.io/cli`. The top navigation bar includes the Angular logo, links for FEATURES, DOCS, RESOURCES, EVENTS, and BLOG, a search bar, and social media icons. The left sidebar lists the documentation structure: GETTING STARTED, TUTORIAL, FUNDAMENTALS, TECHNIQUES, SETUP & DEPLOYMENT, RELEASE INFORMATION, QUICK REFERENCE, and CLI COMMANDS (which is expanded to show Overview, ng add, ng build, ng config, ng doc, ng e2e, ng generate, ng help, ng lint, and ng new). The main content area is titled "CLI Command Reference" and includes an introduction to the Angular CLI, a section on "Installing Angular CLI" with a code block for `npm install -g @angular/cli`, a link to the GitHub releases page, a "Basic workflow" section, and a code block for `ng help`.

← → ↻ 🏠 `https://angular.io/cli` ☆ [Icons] | [Profile]

**ANGULAR** FEATURES DOCS RESOURCES EVENTS BLOG Search [Twitter]

GETTING STARTED  
TUTORIAL >  
FUNDAMENTALS >  
TECHNIQUES >  
SETUP & DEPLOYMENT >  
RELEASE INFORMATION >  
QUICK REFERENCE >  
CLI COMMANDS ✓  
  Overview  
  ng add  
  ng build  
  ng config  
  ng doc  
  ng e2e  
  ng generate  
  ng help  
  ng lint  
  ng new

## CLI Command Reference

The Angular CLI is a command-line interface tool that you use to initialize, develop, scaffold, and maintain Angular applications. You can use the tool directly in a command shell, or indirectly through an interactive UI such as [Angular Console](#).

### Installing Angular CLI

Major versions of Angular CLI follow the supported major version of Angular, but minor versions can be released separately.

Install the CLI using the npm package manager:

```
npm install -g @angular/cli
```

For details about changes between versions, and information about updating from previous releases, see the Releases tab on GitHub:  
<https://github.com/angular/angular-cli/releases>

### Basic workflow

Invoke the tool on the command line through the ng executable. Online help is available on the command line. Enter the following to list commands or options for a given command (such as [generate](#)) with a short description.

```
ng help
```

<https://angular.io/cli>



# Angular 2 Code - Backend

Kort over TypeScript en ES6

# Programmeertalen





A Venn diagram consisting of three concentric circles. The outermost circle is dark teal and contains the text 'TypeScript'. Inside it is a medium teal circle containing the text 'ES6'. Inside the 'ES6' circle is a light teal circle containing the text 'ES5'. This visualizes that TypeScript is a superset of ES6, and ES6 is a superset of ES5.

TypeScript

ES6

ES5



# ES6 en TypeScript

**De toekomst van JavaScript is ES6/ES2015**

Major update van JavaScript als programmeertaal

Modules, classes en meer

Helpt bij het ontwikkelen in Angular 2

**TypeScript breidt ES6 verder uit**

Annotaties & types

Interfaces

Compiler

## **TypeScript – tooling support**

Types, Autocompletion.

Compile-time checking in editors.

Alles is *optioneel*. Je kunt altijd nog gewoon JavaScript gebruiken.

# Onderdelen van een Component Class

imports

```
import { Component } from '@angular/core';  
import { DataService } from '../services/data-service';
```

annotations

```
@Component({  
  selector: 'orders',  
  directives: [DataService],  
  templateUrl: 'orders-component.html',  
})
```

class

```
export class OrdersComponent {  
  ...  
}
```

# Checkpoint

- Angular 2 is een totaal ander framework dan Angular 1
- Component-based vs. Page-based
- Nieuwe syntaxis
- Nieuwe programmeertalen en andere nieuwe kenmerken
- Concepten komen – grotendeels – overeen
- Veel boilerplate-code nodig voor een Quickstart
- Daarna: niet meer naar omkijken. Concentreren op de componenten